Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

CuikSLAM: A Kinematics-based Approach to SLAM

Josep M. Porta
Institut de Robòtica i Informàtica Industrial (UPC-CSIC)
Llorens i Artigas 4-6, 08028 Barcelona
porta@iri.upc.edu

***Abstract*— In this paper, we depart from the fact that Simultaneous Localization and Mapping (SLAM) is a sub-case of the general kinematic problem, and, thus, all techniques used in kinematics are potentially applicable to SLAM. We describe how to formalize a SLAM problem as a typical kinematic problem and we propose a simple SLAM algorithm based on an interval-based kinematic method called Cuik previously developed in our group. This new algorithm solves the SLAM problem taking advantage of the structure imposed in the SLAM problem by the motion and sensing capabilities of the autonomous robots. However, since we use a kinematic approach instead of a probabilistic one (the usual approach for SLAM) we can perfectly model the constraints between robot poses and between robot poses and landmarks, including the nonlinearities, and we can ensure those constraints to be fulfilled at any time during the map construction and refinement. The viability of the new algorithm is shown with a small test.**

***Index Terms*— SLAM, Kinematics, Interval-based methods.**

I. Introduction

The objective of kinematics is to find all the possible placement of a set of objects that fulfill a given set of *kinematic constraints* [1]. For instance, in a robot arm the constraints define the valid range of movements of each motor, and the desired pose for the robot's gripper. In this case, the objective is to find all the possible positions for the motors so that the desired pose for the gripper is achieved. However, the generality of the kinematics's objectives makes kinematics's methods valid not only for robotics [2] but also for disciplines such as computer graphics, CAD [3] or even molecular biology [4].

In the recent years, the robotic community has payed large attention to the so called Simultaneous Localization and Mapping (SLAM) problem [5]. An autonomous robot performing SLAM must define a map of its surroundings from its sensor readings and simultaneously use this under-construction map to find out its position in the environment. In the most popular approach to SLAM, the map built by the robot includes a set of landmarks, i.e., distinctive sensor readings that the robot can detect and re-identify in different moments. Usually SLAM algorithms define an *absolute map* [6] where landmarks poses are defined with respect to a pre-defined frame of reference (most usually, the initial pose of the robot). Since the external frame of reference can be arbitrarily changed, the real information contained in the map is basically the relative position between landmarks, i.e., a *relative map* [6]. Moreover, the position of the robot at a given time can be readily determined if we know its relative pose with respect to a sub-set of the landmarks in the map. In other words, the objective of SLAM can be seen as that of finding the possible relative placements between a set of objects (i.e., the robot and the landmarks). With this reformulation, it is clear that SLAM is a sub-case of the more general kinematic problem.

Despite this clear parallelism, methods used to solve the SLAM problem largely differ from those used in kinematics.

Classical SLAM methods [7] rapidly converge to *one* solution, i.e., the most likely distribution of landmarks and robot poses according to the observations made by the robot. When sampling techniques are used [8] the number of samples must be large if we want to ensure that valid solutions are not discarded and large sample sets make these methods computationally expensive. In any case, using probabilistic techniques there is a chance that maps and robot's poses with low likelihood, although possible, are discarded. This is in contrast with the objective of kinematic methods that is to isolate *all* the possible solutions for a given set of kinematic constraints. In many cases, especially when little information is available about the pose of a given object, more than one solution exists and picking one of them, even if it is the most likely one, can be a rather arbitrary and a source of problems in later stages of the mapping process [9].

Additionally, in the probabilistic approach the uncertainty in the placements between two objects is usually represented using simple functions such as Gaussians defined in the physical space where the robot moves. However, in general, the restriction between two objects (robot poses, landmarks, etc.) define more complex shapes. Therefore, since the assumed probability distributions are just approximations, the single solution obtained via probabilistic methods is just an approximation to the actual most likely solution. In many cases, the approximation found via probabilistic methods does not even fulfill the kinematic constraints (i.e., it is not included in the set of valid poses between two objects) and special correction procedures must be devised [6]. In kinematics, this problem is avoided by working in the *configuration space* (the space of the degrees of freedom between objects). In this space, the valid relative poses between objects are represented as simple axis-aligned n-dimensional boxes and, therefore, no approximation is necessary. Some SLAM methods also work in the configuration space [10], [11]. These techniques use optimization processes to find the solution to the mapping problem that maximizes a given likelihood

0-7803-8914-X/05/$20.00 ©2005 IEEE.

(or energy) function. These SLAM methods always find a valid solution, i.e., a solution that fulfills the kinematic constraints, but this solution is not guaranteed to be the global optimal one since, due to the non-linearities in the constraints, they can easily get trapped at local minima.

Finally, usual SLAM algorithms do not explicitly exploit the structure of the mapping problem (connections between the poses of the robot, and between robot's poses and landmarks, etc.). For instance, the covariance matrix used in the popular Kalman approach to SLAM includes information about the pose correlation between all elements in the scene, including the robot. Some of the pioneer works in SLAM [6], [12] and also some recent works [10], [13], however, propose to exploit the structure of the mapping problem as a key issue to obtain efficient solutions to the SLAM problem. The idea of exploiting the connection between elements in the problem is in the kernel of kinematics.

In this paper, we go a bit further in the line of applying typical kinematic principles and methods to SLAM, in particular, interval-based methods. The advantage is that, using interval-based methods, all valid solutions to the SLAM problem are isolated and we avoid the problem of local minima that affect optimization-based techniques.

In Section II, we formulate the SLAM problem as a set of kinematic constraints. Departing from this result, any general kinematic solver can be applied to the SLAM problem. In Section III, we apply Cuik, an interval-based kinematic solver previously developed in our group, to the SLAM problem. Interval methods have been previously applied to robot localization [14] and to SLAM [15] with the objective of defining absolute maps. In our approach we take a kinematic approach and, thus, we define relative maps by working in the configuration space. The advantages of doing so in terms of memory usage and map update time have been outlined by several authors before [12]. In Section IV, we validate the CuikSLAM algorithm with a simple example and, in Section V, we summarize our work and we point to lines for further interchange of ideas and methods between kinematics and SLAM.

II. SLAM as a Kinematic Problem

In kinematics, the relative pose of one object with respect to another is modeled as a *kinematic link*. In an Euclidean 3D space, a general kinematic link between objects A and B has must include 6 degrees of freedom (3 translations and 3 rotations). So, a general link can be, for instance,

$$\begin{aligned} L_{A,B} =& T_x(x_{A,B})T_y(y_{A,B})T_z(z_{A,B}) \\ & R_x(\theta_{A,B})R_y(\phi_{A,B})R_z(\psi_{A,B}), \end{aligned}$$

with T_i a translations along axis i, R_i a rotation about axis i, and $(x_{A,B}, y_{A,B}, z_{A,B}, \theta_{A,B}, \phi_{A,B}, \psi_{A,B})$ a set of parameters for the transforms. A typical way to represent these transforms is using 4×4 homogeneous matrices. Some of the parameters of the link are fixed, but others are variables with valid values in given ranges. A set of objects and the kinematic links between them define a *kinematic graph* [16]. In general, a graph includes cycles composed by a sub-set of objects sequentially connected by kinematic links. Each cycle in a kinematic graph define a *kinematic equation*

$$L_{1,2}L_{2,3}L_{3,4}\dots L_{N,1} = I$$

with I the 4×4 identity matrix. The number of cycles in a graph can be rather large, but it is enough to work with a base of the cycles of the graph since the rest of cycles define equations that are linear combinations of those in the base. A solution to the kinematic problem is an assignment of variables to values so that all kinematic equations are fulfilled.

The relation between two consecutive poses of an autonomous robot moving on a planar surface can be modeled as a kinematic link with three parameters

$$L_{s_t,s_{t+1}} = T_x(x_{A,B})T_y(y_{A,B})T_z(0)R_x(0)R_y(0)R_z(\psi_{A,B}).$$

The valid ranges for the parameters $(x_{A,B}, y_{A,B}, \psi_{A,B})$ can be inferred from odometry and the motion error model. Since we only have translations along x and y and rotation about z, we can just use 3×3 homogeneous matrices to represent the transforms. Thus, the above equation reads to

$$\begin{aligned} L_{s_t,s_{t+1}} = & \begin{pmatrix} 1 & 0 & x_{A,B} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & y_{A,B} \\ 0 & 0 & 1 \end{pmatrix} \\ & \times \begin{pmatrix} \cos\psi_{A,B} & -\sin\psi_{A,B} & 0 \\ \sin\psi_{A,B} & \cos\psi_{A,B} & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

As the robot moves, it detects landmarks. The relation between the robot and a landmark observed at a given time can also be modeled as a kinematic link with parameters inferred from the sensor readings and the observation error model. Therefore, at a given time slice we can add to a pre-defined kinematic graph one node representing the current robot's pose, a link from this node to the node representing the previous pose and a different link from the new node to each one of the landmarks detected from this pose (some of the landmarks will be new nodes and others could be nodes already in the graph). This simple incremental procedure allows us to maintain an up-to date kinematic graph representing the SLAM problem at hand. This approach is very close to that already presented in [6]. The main difference of our approach w.r.t. the work presented in [6] is the specific technique used to process the graph in order to determine the map and the robot's location.

Using a kinematic graph, the pose of a given object A (the robot at time t, a landmark, etc.) with respect to the initial pose of the robot, n_1 (that we take as the global reference frame) can be computed as

$$P_A = L_{n_1,n_2}\dots L_{n_k,A}.$$

Observe that, in general, there exist more than one path from n_1 to A. To find P_A as accurately as possible, we should select the shortest path (shortest in the sense of less error and not of less number of kinematic links).

The possible values of the variables in the links encode all the valid maps and robot locations according to the information collected by the robot up to a given moment. Ideally we would like to determine a single map and robot location, that is, to reduce the valid ranges for the variables to single values.

As mentioned, the cycles in kinematic graph define a set of kinematic equations. For instance, a landmark observed from two consecutive poses define a cycle including the landmark and the two robot's poses from which the landmark was observed. This loop defines a kinematic equation that can be used to reduce the ranges of the variables in the loop removing those values for which the cycle equation is not fulfilled (i.e., to remove from the space of possible maps and robot's poses encoded in the kinematic graph those that are not longer valid). Exactly in the same way, when the robot's revisits an area, its path forms a loop and a cycle equation is established that can be used to reduce the ranges of the variables of the links that define the robot's path. Thus, landmarks and robot's path loops can be treated in a unified way since they both define kinematic equations.

Notice that the presence of landmarks is optional, since we can also reduce the error in robot's pose just using the equations derived from the cycles in the robot's path. In this way, the kinematic-based approach unifies the landmark-based SLAM with the map construction techniques based on consistent pose estimation [11]. Note, however, that the detection of landmarks define cycles in the kinematic graph before the robot closes a loop. This landmark-based cycles allow us to keep a low uncertainty on the robot's pose. Additionally, loops formed by the landmarks are likely to be short since landmarks are likely to be detected from close poses and, in general, the shorter the loops, the stronger the constraints and the larger the reduction of the link variables involved in the loop equation.

III. CUIKSLAM

Several approaches have been followed in an attempt to solve the general kinematic problem. Actually, since kinematic problem can be posed as a set of equations, any method to find the solutions to a general set equations can be, in principle, used.

While all kinematic methods should be *complete* (they should be able to find *all* solutions) and *general* (they should be able to tackle *any* system of multivariate polynomial equations), typical kinematic method such as algebraic-geometric methods [17], [18] or continuation methods [19], [20] present some limitations in practice. Probably the most important one is that neither of these approaches is able to obtain the solution variety, or at least characterize it to some extent, if its dimension is greater than zero as it happens, for instance, in redundant mechanisms (and most likely in many SLAM problems).

Interval-based [21]–[23] and subdivision [24], [25] techniques are numerical methods that can discretize the solution variety for an arbitrary mechanism, even if this variety is multidimensional. Additionally, these methods are numerically stable and rather easy to implement. Finally, they can deal with the equations in their input form, thus avoiding the need of intuition-guided symbolic reductions. This is a critical point when applying kinematic methods to SLAM since in SLAM the kinematic graph (and, thus, the kinematic equation set) is on-line extended requiring the solver to be completely autonomous.

Cuik-0 is the first interval-based general kinematic solver developed in our group some years ago [23]. This algorithm has two steps. In the first one, we identify the cycles in the kinematic graph. In the second one, we process the equations derived from each loop with the objective of reducing as much as possible the ranges of the involved variables. Next we describe these two procedures.

To derive the kinematic equations from the kinematic graph representing the SLAM problem at hand, we have to find a base of the cycles of the graph. This can be done using a width-first expansion three of the graph, that can be initialized with just the root node (the initial pose of the robot) an extended as the robot moves using Prim's algorithm.

The expansion tree only includes a sub-set of all graph edges. Each graph edge not included in the expansion three closes a cycle that belongs to the base of cycles we are seeking for. The cycle established by an edge between leaves of the expanding tree a and b, includes all the nodes in the tree from node c to both a and b (including c), where c is the deepest node in the tree common to the paths from the root to a and b.

As the kinematic graph corresponding to a given SLAM problem can be built on-line, so can be the expansion three and the basis of cycles to be considered by Cuik.

In general, we prefer short cycles to longer ones since, as mentioned, short cycles tend to be more restrictive. The detection of cycles using a width-first expansion tree generates short cycles but, to ensure that we detect the cycle base with the shortest longest cycle, a more sophisticated algorithm should be used [26].

The second phase of the Cuik algorithm is to process each one of the kinematic equations derived from the cycles established by the graph edges added at the current time slice. The version of Cuik we use here is based on interval arithmetics that is an extension of real arithmetics where operations are defined on intervals and not on scalars. For instance, for a couple of intervals $a = [\underline{a}, \overline{a}]$ and $b = [\underline{b}, \overline{b}]$ we have that

$$a + b = [\underline{a} + \underline{b}, \overline{a} + \overline{b}].$$

In a similar way we can define all operations that appear in kinematic equations, including the trigonometric functions.

Using interval arithmetics, a given kinematic equation with ranges for the variables can be directly evaluated yielding an interval matrix. This interval matrix must include the identity since, otherwise, the loop can not be closed meaning that there is no solution in the given variable ranges. However, in the SLAM process, the kinematic equation must have at least one solution provided the action and sensor errors are correctly modeled. We could analyze sub-ranges of each variable range discarding those that do

not fulfill the above condition. However, more sophisticated ways to reduce the variable ranges can be devised.

A kinematic equation can be divided in its rotation part and it translation one. The rotation equation includes only the rotation transforms that appear in the kinematic equation. The meaning of this equation is just that the final orientation of the loop must be the same as that at the initial element. Thus, the rotation equation can be see as

$$\sum \theta_i = 0$$

for θ_i the rotation variables included in the kinematic equation. Therefore, for a given interval θ_j we can possibly reduce its range using the following update rule

$$\theta_j \leftarrow \theta_j \cap \Big[-\sum_{i \neq j} \theta_i \Big],$$

that can be applied for all rotational variables in the equation.

The translation part of the kinematic equation is not as simple as the rotational one since it includes both translational and rotational variables. For a given translation along X we can see the kinematic equation as

$$\begin{array}{ccc} \underbrace{M_1 \dots}_{A} & T_x(x) & \underbrace{\dots M_n}_{B} = I \\ & T_x(x) & \end{array} \tag{1}$$

with M_i an interval homogeneous transforms corresponding to an individual degree of freedom and A and B interval homogeneous transforms that represent poses in a plane. The above equation can be made explicit

$$\begin{pmatrix} a_c & -a_s & a_x \\ a_s & a_c & a_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_c & -b_s & b_x \\ b_s & b_c & b_y \\ 0 & 0 & 1 \end{pmatrix} = I$$

From this, we can define two equation involving x

$$\begin{aligned} a_c b_x - a_s b_y + a_c x + a_x = 0 \\ a_s b_x + a_c b_y + a_s x + a_y = 0, \end{aligned}$$

and we have the following update rule for the range of x

$$x \leftarrow x \cap \frac{-a_c b_x + a_s b_y - a_x}{a_c} \cap \frac{-a_s b_x - a_c b_y - a_y}{a_s}.$$

With a similar reasoning, the update rule for ranges of variables associated with translations along Y is

$$y \leftarrow y \cap \frac{a_c b_x - a_s b_y - a_x}{a_s} \cap \frac{-a_s b_x - a_c b_y - a_y}{a_c}.$$

Since the translation equation also involves rotational variables, a similar process can be followed to use the translation equation to reduce rotation variable ranges. The update rule in this case is

$$\begin{aligned} \theta \leftarrow \theta \cap \arctan \frac{-a_s b_c - a_c b_s}{a_c b_c - a_s b_s} \\ \cap \arctan \frac{a_c(b_x a_x + a_y b_y) + a_s(a_y b_x - b_y a_x)}{a_s(b_x a_x + a_y b_y) - a_c(a_y b_x - b_y a_x)}. \end{aligned}$$

In a kinematic equation, the initial frame of reference can be arbitrarily attached to any of the objects in the kinematic cycle. Thus, for a cycle with N links we can define N equations and each one of them produces a different A and B matrices (see Eq. 1) and, thus, a different update rule for each variable.

All of the above update rules can be used for a given kinematic equation as long as there is a variable that shrinks. If not variable range can be reduced, we proceed with the another equation. If, in this process, we manage to reduce the ranges for variables of a edge shared with another cycle, then this cycle equation has to be added to the list of equations to be processed. In this way, we can back-propagate the information obtained at a given time to previous robot and landmarks poses. The process stops when no variable can be reduced with any kinematic equation.

In the original Cuik algorithm, if the ranges for the variables can not be reduced any more, bisection of one of the variables is used to produce two sub-problems that are recursively processed using the procedure just described. Bisection allows us to discretize the solution varieties with an arbitrary resolution and to detect separate solutions. However, we assume the SLAM problem to have a single solution in the limit (i.e., when enough information is available) and, thus, we do not use bisection in CuikSLAM.

A problem of interval arithmetics is that, in many cases, operations produce an overestimation of the final result. This happens when we evaluate an expression that includes correlated sub-expression. For instance, the evaluation of $10x - 8x$ for $x = [1, 5]$ should result in the interval $[2, 10]$ but, using simple interval arithmetics, the result is $[-30, 42]$, since the two x in the expression are assumed as independent variables when they are not. The tighter the evaluation of matrices A and B in Eq. 1 the larger the reduction produced by the above update rules and, thus, it is desirable to minimize this overestimation as much as possible. One possibility to reduce this effect is to use affine arithmetics [27] that automatically takes into account some of the the relations between sub-expressions. However its implementation is not as simple as that of plain interval arithmetics. In the version of Cuik we are using here, we introduced a couple of special operations to minimize the overestimation. First, while a sequence of interval matrix products is evaluated, we have to ensure that any intermediate resulting matrix M fulfills the following conditions

$$\begin{aligned} M_{1,1}, M_{1,2}, M_{2,1}, M_{2,2} \in [-1, 1] \\ M_{1,1} = M_{2,2} \\ M_{1,2} = -M_{2,1} \\ {M_{1,1}}^2 + M_{2,1}^2 = 1 \end{aligned}$$

with $M_{i,j}$ the i, j-th element of matrix M. The application of these rules reduces overestimation while interval matrix products are evaluated, reducing the final overestimation. Another special operation introduced by Castellet [28] is the tight evaluation of expressions such as

$$m \cos\theta + n \sin\theta$$

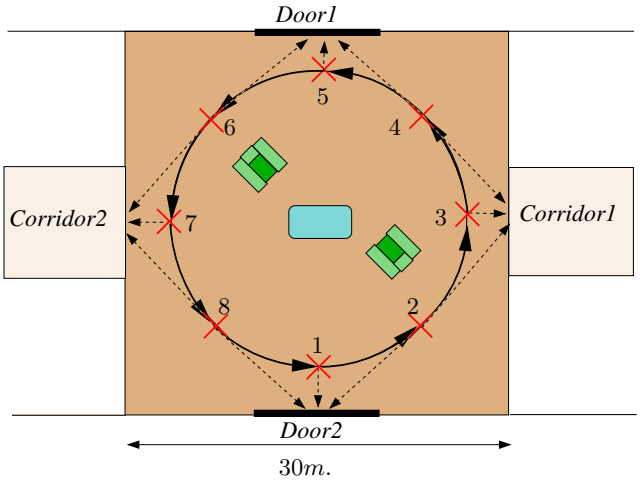

Fig. 1. Map of the test environment. Solid arrow indicate the path of the robot and dashed ones are the visible landmarks from each robot's pose.

with m, n and θ intervals. A careful analysis of this particular expression, that is frequent in homogeneous matrix product, allows us to cancel the overestimation that otherwise would be produced by the double use of θ in the expressions.

IV. Example

To test the feasibility of the proposed kinematic-based SLAM algorithm, we conducted a simple simulated test. In this test (see Fig. 1), the robot moves on a 20 meters diameter circle on a 30×30 meters lobby. The robot is equipped with a ring of sonars from which it can detect large openings in the walls like corridors of doors. Those openings are used as landmarks with a frame of reference attached to the center of the initial and final points of the opening. The path of the robot is loop composed by 8 different poses departing from pose 1 that is the root of the expansion tree and whose frame of reference is taken the global one. From each pose only the closest doors/corridors can be perceived. The set of visible landmarks from each pose is depicted in Fig. 1.

In the displacement from one robot's pose to another, robot odometry is affected by a maximum noise of ± 25 cm. in X and Y and ± 5 degrees. The error in the sensor readings is 10 times smaller than the error in odometry. These odometry error values correspond to those obtained with a Nomad Scout robot.

The final kinematic graph and the kinematic cycles associated with this example are depicted in Fig. 2.

Fig. 3 shows the X-Y error when only odometry is used to estimate the robot's pose (red rectangles centered on the dotted circle that represents the path of the robot) and the landmarks poses (blue rectangles not centered on the circle). Rectangles in solid line correspond to the estimation using plain interval arithmetics and rectangles in dashed line to the result of using the overestimation reduction procedures described at the end of Section III. These special procedures reduce the uncertainty of the

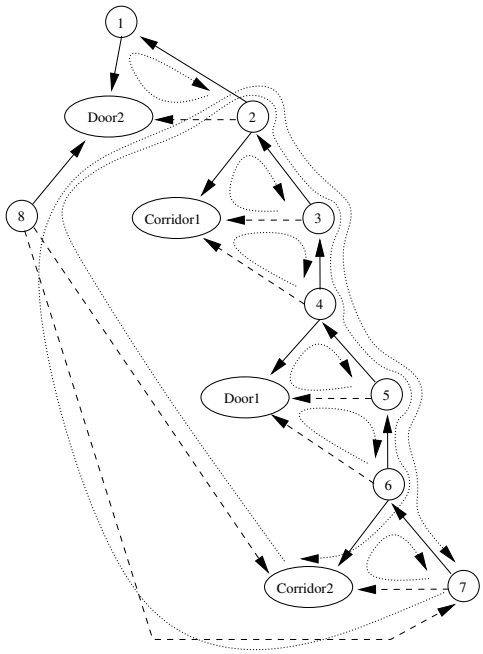

Fig. 2. Expansion tree corresponding to the kinematic graph derived from the example in Fig. 1. Solid arrow are edges in the expansion tree, dashed arrows are edges that establish a cycles and dotted lines indicate the cycles.

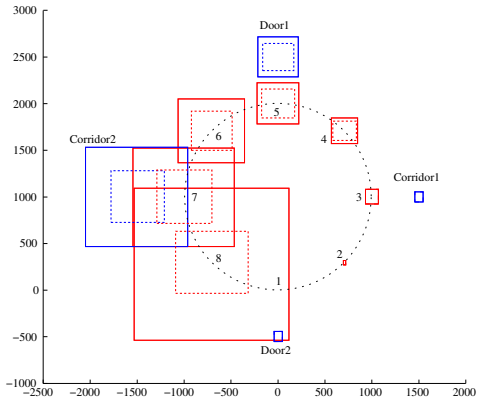

Fig. 3. Poses of the robot (red) and landmarks (blue) estimated using odometry. Units are centimeters.

robots and landmarks poses. This error reduction effect is specially noticeable in robot's pose 8 and it would be larger for longer chains of links (i.e., for longer robot paths).

Fig. 4 shows the error reduction achieved when the robot closes its path (i.e., re-detects landmark *Door2*). Solid and dashed lines are the poses estimated before and after closing the loop, respectively. The effect of closing the robot's path is a large reduction of the error of robot's poses 6, 7 and 8. This reduction is caused by two effects. The first one is a small reduction in the variables of the links representing the robot path achieved by processing the kinematic equation established when the robot path is closed. The second and more significant error reduction is achieved by the possibility of computing the poses for points 6, 7 and 8 using the path $\{$*1, Door2, 8, 7, 6*$\}$ that is

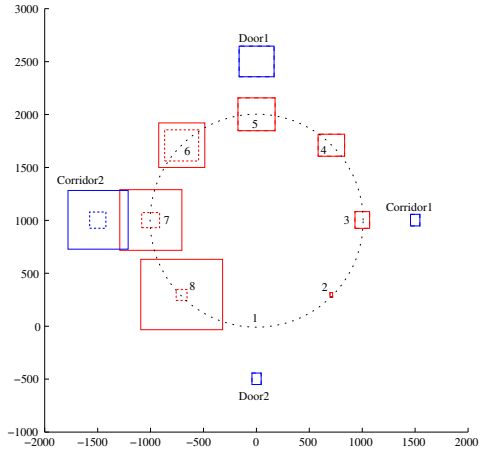

Fig. 4. Effect on the robot and landmarks pose estimations when closing the robot path.

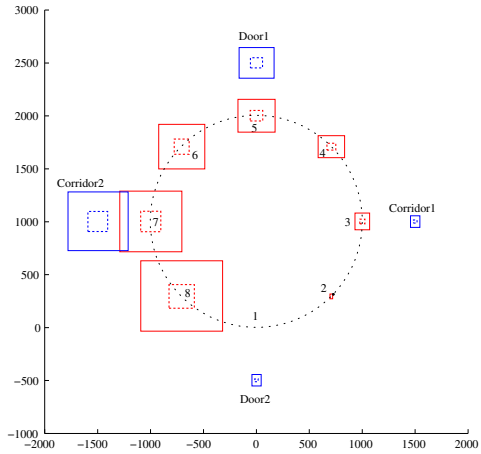

Fig. 5. Robot and landmarks pose estimations using landmarks in an open path.

shorter (in the sense of less error) than the path from 1 to 8 that was the only available path before closing the loop.

Fig. 5 shows the pose estimation error taking advantage of the landmarks when the robot follows a path from 1 to 8 without re-detecting *Door2* from 8. Landmarks observed as the robot moves form local kinematic cycles (see Fig. 2) that are used to reduce the variables in the kinematic links. This on-line reduction allows to get an estimation of the final pose of the robot (the pose at point 8) with much less error than that obtained using only odometry (the solid rectangles in the figure).

Finally, Fig. 6 shows the combined effect of using the local cycles established by the robot's path and the landmarks and the global cycle established when the robot re-detects *Door2*. In this case, the larger error is at point 5 that is below 1 meter. This is quite small compared with the size of the environment and with the 4-meter error obtained when using only odometry (see Fig. 3). The poses of passing points and landmarks close to the initial pose of the robot can be determined with high accuracy: below 15 cm for point 8. The error for this point pose estimation when

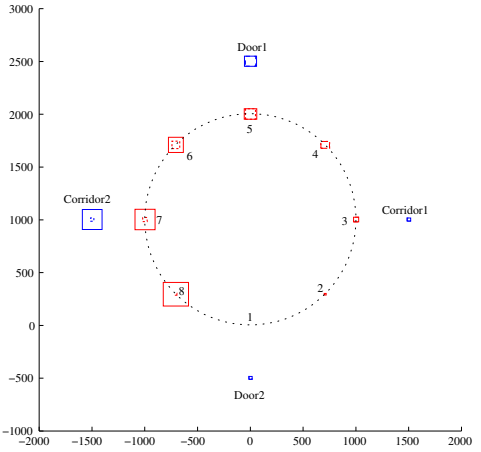

Fig. 6. Robot and landmarks pose estimations using landmarks in a closed path.

using only odometry and not closing any loop was above 7 meters. Thus, even though its simplicity, CuikSLAM allow for a significant error reduction for the landmarks poses (i.e., the map) and for the robot's location.

V. Conclusions

In this paper, we outlined the close relation between kinematics and SLAM by posing SLAM as a particular case of the general kinematic problem. The consequence is that we can readily apply typical kinematic methods to SLAM. These methods take advantage of the underlying structure of the SLAM problem, as it is also done some of the existing SLAM algorithms [6], [9], [10], [12], [13]. In this line, we introduced CuikSLAM, the use of a general interval-based kinematic solver to SLAM. CuikSLAM is a conservative algorithm in the sense that it never discards any valid solution for the problem at hand. Thus, our SLAM method does not suffer from the problem derived from selecting just one of all the possible solutions (not always the global optimal one) that affects many of the existing methods. The main inconvenient of interval-based methods is the overestimation effect. This effect makes our solver to consider as solutions some points that are actually not valid solutions. To avoid the overestimation we have to use more sophisticated (and slower) interval-based solvers [25]. However, the overestimation reduction rules introduced in this paper are a good trade off since they represent a small computation overhead, but they largely reduce overestimation.

The parallelism between SLAM and kinematics is complete: any planar multi-loop mechanism can be seen as the path of a robot on a planar surface (possibly with landmark observations) and, the other way around, any SLAM problem can be seen as a mechanism. The advantage of this parallelism is that methods from kinematics can be directly translated to SLAM (as done in this paper) and, the other way around, SLAM methods could potentially be used in kinematics. The bad news is that the general kinematic problem is NP-complete and, consequently, so

is the general SLAM problem (however, this is not the case when simplifying assumptions such as the Markov property are taken in SLAM).

In this paper, we introduced a new SLAM approach and proved the feasibility of the concept. However much work must be done to implement the ideas introduced here in a real robot. For instance, in this paper we assumed the data association problem (recognition of landmarks) as solved. Alternative associations produce different kinematic graphs, but only the correct association would produce a consistent graph (a graph with solutions). Therefore, one possibility to deal with the data association problem is to track all possible graphs (i.e., all possible data associations) and to eliminate the ones that, at a given moment, are proved to be inconsistent.

Several other lines for further work should be considered. For instance, it would be interesting to apply to SLAM distance geometry solvers [29], [30] specially aimed to deal with large kinematic problems such as those that appear in hyper-redundant robots [31] or in protein folding [32]. Rigidity theory [33] can be used to determine which edges to be added to the graph to reduce the mobility of the mechanism or, in other words, to determine which robot movements to be executed to reduce the number of possible maps and robot poses.

ACKNOWLEDGMENTS

I would like to express my gratitude to Prof. Federico Thomas and Dr. Lluís Ros for the fruitful discussions during the preparation of this paper.

This work has been partially supported by the Spanish CICYT under contract TIC2004-07358, the Catalan Research Commission through the "Robotics and Control" group, and by a Ramón y Cajal contract from the Spanish Ministry for Science and Technology.

REFERENCES

[1] J. Angeles, *Spatial Kinematic Chains. Analysis, Synthesis, and Optimisation*. Springer-Verlag, Berlin, 1982.

[2] J. Nielsen and B. Roth, "On the kinematic analysis of robotic mechanisms," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1147–1160, 1999.

[3] C. H. Hoffmann and B. Yuan, "On spatial constraints solving approaches," in *Automated Deduction in Geometry: Third International Workshop*, ser. Lecture Notes in Computer Science, vol. 2061, 2000.

[4] I. Z. Emiris and B. Mourrain, "Computer algebra methods for studying and computing molecular conformations," *Algorithmica*, no. 25, pp. 372–402, 1999.

[5] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2003.

[6] P. Newman, "On the structure and solution of the simultaneous localization and map building problem," Ph.D. dissertation, Australian Center of Field Robotics, University of Sydney, 2000.

[7] J. Leonard and H. Durrant-Whyte, "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[8] A. Doucet, N. d. Freitas, and N. Gordon, *Sequential Monte Carlo in Practice*. Springer-Verlag, New York, 2001.

[9] J. Folkesson and H. Christensen, "Graphical SLAM: A self-correcting map," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 383– 390.

[10] K. Konolige, "Large-scale map-making," in *AAAI*, 2004, pp. 457–463.

[11] F. Lu and E. Milios, "Globally consistent range scan aligment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[12] M. Csorba, "Simultaneous localization and map building," Ph.D. dissertation, Robotics Research Group, Department of Engineering Science, University of Oxford, 1997.

[13] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[14] M. Kieffer, L. Jaulin, E. Walter, and D. Meizel, "Robust autonomous robot localization using interval analysis," *Reliable Computing*, vol. 6, no. 3, pp. 337–362, 2000.

[15] M. D. Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "A set theoretic approach to dynamic robot localization and mapping," *Autonomous Robots*, vol. 14, pp. 23–47, 2004.

[16] F. Thomas, *Computer-Aided Mechanical Assembly Planning*. Kluwer Academic Publishers, 1991, ch. Chapter 4: Graphs of Kinematic Constraints, pp. 81–111.

[17] D. Manocha and J. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 648–657, 1994.

[18] K. Sridharan, "Computing two penetration measures for curved 2D objects," *Information Processing Letters*, vol. 72, pp. 143–148, 1999.

[19] B. Roth and F. Freudenstein, "Synthesis of path-generating mechanisms by numerical methods," *ASME Journal of Engineering for Industry*, vol. 85, pp. 298–307, 1963.

[20] C. Wampler, A. Morgan, and A. Sommese, "Numerical continuation methods for solving polynomial systems arising in kinematics," *ASME Journal of Mechanical Design*, vol. 112, pp. 59–68, 1990.

[21] R. S. Rao, A. Asaithambi, and S. K. Agrawal, "Inverse kinematic solution of robot manipulators using interval analysis," *ASME Journal of Mechanical Design*, vol. 120, pp. 147–150, 1998.

[22] J.-P. Merlet, "A formal numerical approach to determine the presence of singularity within the workspace of a parallel robot," in *Proceedings of the 2nd Workshop on Computational Kinematics*, Seoul, South Korea, May 2001, pp. 167–176.

[23] A. Castellet and F. Thomas, "An algorithm for the solution of inverse kinematics problems based on an interval method," in *Advances in Robot Kinematics*, M. Husty and J. Lenarcic, Eds. Kluwer Academic Publishers, 1998, pp. 393–403.

[24] E. Sherbrooke and N. Patrikalakis, "Computation of the solutions of nonlinear polynomial systems," *Computer Aided Geometric Design*, vol. 10, no. 5, pp. 379–405, 1993.

[25] J. M. Porta, L. Ros, F. Thomas, and C. Torras, "Solving multiloop linkages by iterating 2D clippings," in *Advances in Robot Kinematics*, F. Thomas and J. Lenarcic, Eds. Kluwer Academic Publishers, 2002, pp. 255–264.

[26] P. Gleiss, J. Leydold, and P. Stadler, "Interchangeability of relevant cycles in graphs," *Electron. J. Combin.*, pp. 1–16, 2000.

[27] J. Stolfi and L. de Figueiredo, *Self-Validated Numerical Methods and Applications*. Rio de Janeiro: Institute for Pure and Applied Mathematics (IMPA), 1997, monograph for the 21st Brazilian Mathematics Colloquium (CBM'97), IMPA.

[28] A. Castellet, "An interval method for solving inverse kinematics problems," Ph.D. dissertation, Universitat Politècnica de Catalunya, 1998.

[29] J. M. Porta, L. Ros, F. Thomas, and C. Torras, "A branch-and-prune solver for distance constraints," *IEEE Transactions on Robotics*, p. To appear.

[30] F. Thomas, "Solving geometric constraints by iterative projections and backprojections," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 1789–1795.

[31] G. Chirikjian and J. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343–354, 1994.

[32] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation*. Research Studies Press: Taunton, Somerset, England, 1998.

[33] M. Thorpe and P. Duxbury, Eds., *Rigidity Theory and Applications*, ser. Fundamental Materials Research. Kluwer, 1999.